Using Minimum-Surface Bodies for Iteration Space Partitioning

*Michael Frumkin and Rob F. Van der Wijngaart**

Abstract

A number of known techniques for improving cache performance in scientific computations involve the reordering of the iteration space. Some of these reorderings can be considered as coverings of the iteration space with the sets having good surface-to-volume ratio. Use of such sets reduces the number of cache misses in computations of local operators having the iteration space as a domain. We study coverings of iteration spaces represented by structured and unstructured grids. For structured grids we introduce a covering based on successive minima tiles of the interference lattice of the grid. We show that the covering has good surface-to-volume ratio and present a computer experiment showing actual reduction of the cache misses achieved by using these tiles. For unstructured grids no cache efficient covering can be guaranteed. We present a triangulation of a 3-dimensional cube such that any local operator on the corresponding grid has significantly larger number of cache misses than a similar operator on a structured grid.

1 Introduction

A number of known techniques for improving cache performance in scientific computations involve the reordering of the iteration space. We present two new methods for partitioning the iteration space with minimum-surface cache fitting sets. Such

*Computer Sciences Corporation; M/S T27A-2, NASA Ames Research Center, Moffett Field, CA 94035-1000; e-mail: {frumkin,wijngaar}@nas.nasa.gov Numerical Aerospace Simulation Systems Division NASA Ames Research Center

partitionings reduce the number of cache misses to a level that is close to the theoretical minimum. We show that the coverings reduce the number of the misses by actual measurements of cache misses in computations of a second order stencil operator on structured three-dimensional grids.

A good tiling of the iteration space for structured discretization grids can be constructed by using the interference lattice of the grid. This lattice is a set of grid indices mapped into the same word in the cache or, equivalently, a set of solutions of the Cache Miss Equation [4]. In [2] we introduced a (generally skewed) tiling of the iteration space of the explicit operators on structured grids with parallelepipeds built on a reduced basis of the interference lattice. We showed that for lattices whose second shortest vector is relatively long the tiling reduces the number of cache misses to a value close to the theoretical lower bound. Constructing the skewed tiling, however, is a nontrivial task, and involves a significant overhead in testing whether a particular point lies inside the tile. Tiling a three-dimensional grid, for example, requires the determination of 29 integer parameters to construct the loop nest of depth six, and involves a significant branching overhead.

In this paper we introduce two new, more practical coverings of structured grids: a covering with Voronoi cells and a covering with rectilinear parallelepipeds built on the vectors of successive minima of the interference lattice. In lattices with a relatively long shortest vector the cells of both coverings have near-optimal surface-to-volume ratios. Hence, the number of cache misses in the computations tiled with these cells is close to the theoretical minimum derived in [2]. Direct measurements of the cache misses show a significant advantage of the successive minima covering relative to the computations using the canonical loop ordering (maximally optimized by a compiler). On the other hand, we construct an unstructured grid that triangulates a 3-dimensional cube and show that the grid can not be covered with sets having good surface-to-volume ratios. The last result shows that any computation of an explicit local operator on such 3-dimensional grid would suffer larger number of cache misses then a computation of a similar operator on a structured grid of the same size.

2 Cache Usage in Computations of Local Operators

Local operators on the grids. We consider the problem of computing a local explicit operator $q = Ku$ on data defined at the vertices of an undirected graph $G = (V, E)$ which we call grid. Locality of the operator K means that computation of $q(x)$, $x \in V$, involves values of $u(y)$, $y \in V$, where y is at a (graph) distance at most k from x. This k is called the order of K and assumed to be independent of G. K is explicit, meaning that the values of q can be computed in arbitrary order.

The structured and unstructured grids we consider have an explicit or implicit embedding into an Euclidean space. Structured grids are Cartesian products of line graphs, while edges of unstructured grids are defined explicitly by an adjacency matrix. We assume that the maximum vertex degree is independent of the total number of vertices. A grid is called a triangulation of a body B if it can be represented as a 1-dimensional skeleton of a simplicial partition of B.

Cache Model. We consider a single-level, virtual-address-mapped, set-associative data cache memory, see [3]. The cache is organized in a sets of z lines of w words each. Hence, it can be characterized by the parameter triplet (a, z, w), and its size S equals $a \cdot z \cdot w$ words. The cache memory is used as a temporary fast storage of words used for processing. A word at virtual address A is fetched into cache location $(a(A), z(A), w(A))$, where $w(A) = A \bmod w$, $z(A) = (A/w) \bmod z$, and $a(A)$ is determined according to a replacement policy.

The number of cache misses incurred in computation of K depends on the order in which elements of u are stored in the main memory. We assume that for structured grids an element $u(i_1, \ldots, i_d)$ is stored at address $A = i_1 + n_1 i_2 + n_1 n_2 i_3 + \cdots + n_1 \cdots n_{d-1} i_d$, where $n_1, \cdots, n_{d-1}$ are the grid sizes. For unstructured grids we don't assume any particular ordering of the grid points (and, hence, elements of u). Instead we choose an ordering that reduces the number of cache misses.

Replacement loads. A *cache miss* is defined as a request for a word of data that is not present in the cache at the time of the request. A *cache load* is defined as an explicit request for a word of data for which no explicit request has been made previously (a *cold load*), or whose residence in the cache has expired because of a cache load of another word of data into the exact same location in the cache (a *replacement load*). The definitions of cold and replacement loads are analogous to those of cold and replacement cache misses [4], respectively, and if w equals 1 they completely coincide.

Surface-to-volume ratio. One technique for minimization of the number of replacement loads is to cover the grid $G = (V, E)$ with conflict-free sets $V = \bigcup V_i$, $i = 1, \ldots, k$, $|V_i| = S$, that is, sets without vertices mapped to the same location in cache. If we calculate q in all vertices of V_i before calculating it in vertices of V_j, $j > i$, a replacement load can occur only at vertices having neighbors in at least two sets (boundary vertices). We consider only bounded degree graphs, so if we can find a covering with sets having volumes $|V_i|$ close to S and a minimal number of boundary vertices $|\partial V_i|$ (and edges), then the computation of K will have a number of replacement loads close to the minimum. The total partition boundary $\sum_{i=1}^{k} |\partial V_i|$ can be used to obtain a lower bound for the number of replacement loads, as shown in [2], cf. [5].

3 Structured Grids

3.1 Interference Lattice

Interference lattice. Let u be a d-dimensional array defined at the vertices of a structured d-dimensional grid of size $n_1 \cdots n_d$. Let L be a set in the index space of u having the same image in cache as the index $(0, \ldots, 0)$. L is a lattice in the sense that there is a generating set of vectors $\{\mathbf{b}_i\}$, $i = 1, \ldots, d$, such that L is the set of grid points $\{(0, \ldots, 0) + \sum_{i=1}^{d} x_i \mathbf{b}_i \,|\, x_i \in \mathbf{Z}\}$. We call L the *interference lattice* of u. It can be defined as the set of all vectors $(i_1, \ldots, i_d)$ that satisfy the Cache Miss Equation [4]:

$$(i_1 + n_1 i_2 + n_1 n_2 i_3 + \cdots + n_1 \cdots n_{d-1} i_d) \bmod S = 0\,.$$

We will use some geometrical properties of lattices. Let B be a convex body of volume V, symmetrical about the origin. The minimal λ_i such that $\lambda_i B$ contains i linear independent vectors of L is called the i^{th} successive minimum of a lattice L relative to B. A theorem by Minkowski, see [1] (Ch. VIII, Th. V), asserts that

$$\frac{2^d}{d!V} \leq \frac{\prod_{i=1}^{d} \lambda_i}{\det L} \leq \frac{2^d}{V}. \tag{1}$$

Note that the ratios of lattice successive minima relative to the unit cube and to the unit ball can be bounded: $1/d \leq \lambda_i^{cube}/\lambda_i^{ball} \leq d$. If B is a unit cube we call $f = \lambda_d/\lambda_1$ the *eccentricity* of the lattice (not to be confused with eccentricity of a reduced basis, defined in [2], Section 4).

3.2 Successive Minima Tiling

In this section we consider tilings with Voronoi cells and with successive minima parallelepipeds. We show that these tilings have good surface-to-volume ratio if the lattice has a small eccentricity.

In [2] we have introduced a tiling by parallelepipeds built on a reduced-basis of the interference lattice, which decreases the number of the cache misses to a level close to the theoretical lower bound that we also derived. Measurement shows that this tiling has significantly fewer cache loads than a compiler-optimized code. However, it has a high computational cost, since it depends on a significant number of integer parameters (29 integers for a 3D grid), and its implementation scans through a significant number of the grid points to select those suitable for cache conflict-free-computations. This prompts us to consider other tilings.

A Voronoi tiling is a tiling of the grid by completed cell C (Voronoi tile) of the Voronoi diagram. For each lattice point x a Voronoi cell is the set of points which are closer to x than to any other lattice point. All integer points inside each Voronoi cell are mapped into the cache without conflicts. Voronoi cells may not form a tiling since some integer points can be located on a cell boundary. There are many qualitatively equivalent ways to complete the cells to form a tiling. One way is to choose a basis in the space of the lattice and assign an integer point on a cell boundary to the cell whose center has the lexicographically smallest projection on the basis vectors.

In order to estimate the surface-to-volume ratio of C we note that the completed Voronoi cells form a tiling of space. Hence, the volume of C equals the determinant of the lattice, which is equal to S, see [2]. On the other hand, each vertex v of C is equidistant from d lattice points. Let r be that distance. Again, according to the definition of the Voronoi cell, the ball of radius r, centered at v, contains no other lattice points. Hence, C is contained in a ball of radius R, centered at $\mathbf{o}$, where R is a maximal ball of the lattice (a ball of maximum radius containing no lattice points). Thus, the surface area of C is bounded by the surface of the maximal ball, which equals dV_dR^{d-1} where $V_d = \frac{\pi^{d/2}}{\Gamma(1+d/2)}$ is the volume of the unit d-dimensional ball (see [1] Ch. IX.7).

We can bound the radius of the maximal ball R_d by

$$R^2 = R_d^2 \leq 1/4 \sum_{i=1}^{d} \lambda_i^2. \tag{2}$$

This can be proved by induction on d (see Figure 1)

$$R_d^2 \leq (h_d/2)^2 + R_{d-1}^2 \leq (\lambda_d/2)^2 + R_{d-1}^2$$

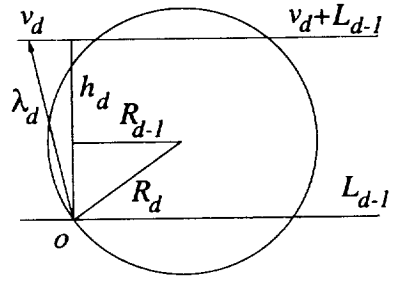

Figure 1. *The radius of maximal ball inscribed into L can be estimated through the radius R_{d-1} of the maximal ball inscribed into the lattice L_{d-1} built on the first $d-1$ minima vectors, and through the value of the last minimum $\lambda_d = |v_d|$. Here h_d is the distance between L_{d-1} and $v_d + L_{d-1}$*

Hence, for the surface area A of C we have the estimation

$$A(C) = dV_d R^{d-1} \leq d(\sqrt{d}/2)^{d-1} V_d \lambda_d{}^{d-1} \leq d^{\frac{d+1}{2}} V_d^{1/d} f^{(d-1)^2/d} S^{(d-1)/d},$$

where we used the estimation $\lambda_d^d \leq \frac{2^d}{V_d} f^{d-1} S$ derived from (1), and the bound $R \leq \frac{\sqrt{d}}{2}\lambda_d$ which follows from (2).

Finally, for the surface-to-volume ratio of the Voronoi cell we have the following estimate:

$$\frac{A(C)}{V(C)} \leq c_d f^{(d-1)^2/d} S^{-1/d}$$

where c_d is a constant depending on d only.

Successive minima tiling. The Voronoi cell tiling has cache-conflict-free tiles of maximum possible volume S, and of good surface-to-volume ratio. However, the tiles may have many faces and it may be computationally expensive to scan through the grid points inside a tile. In this sense it is desirable to use rectilinear tiles. A successive minima tiling is a tiling by a Cartesian block built with use of successive minima lattice vectors of the unit cube. Such a block Q can be described by the system

$$|x_i| \leq b_i, \ i = 1, \ldots, d \tag{3}$$

where $\lambda_1 \leq b_i \leq \lambda_d$.

The block Q can be constructed by the following "inflating" process. Take an initial cube of the form (3) with $b_i = 1$, $i = 1, \ldots, d$, and increment b_i until the face $x_i = b_i$ contains a lattice point. Continue to increment values of all b_j for which the face $x_j = b_j$ has no lattice points. At the end we obtain a block of the form (3) containing a lattice point on each of its faces and containing no lattice points inside except $\mathbf{o}$. In the best case each successive minimum vector will belong to one of the faces of the block, meaning that $b_i = \lambda_i$ (after an appropriate reordering of the coordinates). On the other side, it is not difficult to construct a 3-dimensional lattice such that the block $b_1 = \lambda_1$, $b_2 = b_3 = \lambda_2 < \lambda_3$, so the volume of the block would be strictly less then $\lambda_1 \cdots \lambda_d$.

Any translation of the block $Q' = \frac{1}{2}Q$ obviously contains at most one lattice point and can be used for conflict free tiling. This block has a good surface-to-volume ratio if the lattice has bounded eccentricity, which can be seen from the following inequalities: $A(Q') \leq 2d\lambda_d^{d-1}$ and $V(Q') \geq \lambda_1^d$. Hence, the surface-to-volume of the block can be estimated as follows:

$$\frac{A(Q')}{V(Q')} \leq 2df^{(d-1)^2/d}/\lambda_1 \leq d(d!V_d)^{1/d} f^{d-1} S^{-1/d}$$

since $\lambda_d = f\lambda_1$ and $\lambda_1 \geq 2(\frac{S}{d!V_d})^{1/d} f^{(d-1)/d}$ as follows from (1).

As a representative example, the number of cache misses for tilings of 3-dimensional grids of sizes $40 \leq nx \leq 99$, $ny = 97$, $nz = 99$ with successive minima parallelepipeds is shown in Figure 2. Experiments were performed on an SGI Origin 2000 machine with a MIPS R10000 processor.

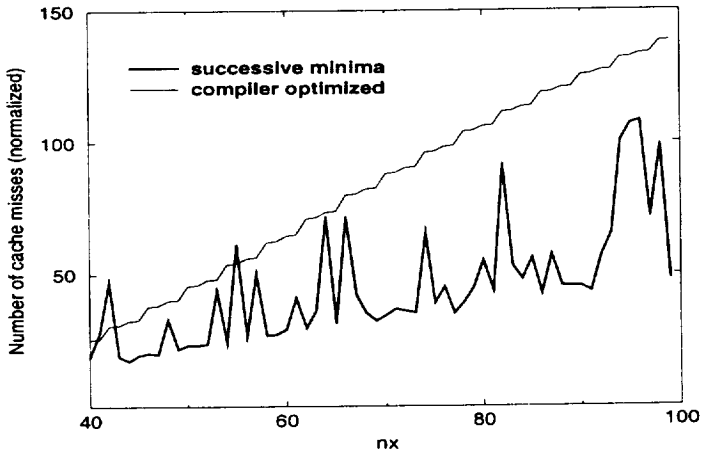

Figure 2. *Comparison of cache misses for a second order stencil operator as a function of the first dimension* ($40 \leq nx \leq 99$, $ny = 97$, $nz = 99$). *The top graph shows the number of cache misses for the compiler optimized nest. The bottom graph is obtained for tilings with successive minima parallelepipeds.*

4 Unstructured Grids – Cache Unfriendly 3-dimensional Grid

In this section we construct an unstructured, bounded-degree 3-dimensional grid of M vertices which has a subgrid of G the size $c_d M$ that does not have small subsets with good perimeter-to-volume ratio. From this property, following the arguments of [2], it can be shown that for any computation of an explicit operator defined on the grid $\Omega(M/logM)$ replacement loads must occur. This shows that if gauged by the number of cache misses, unstructured girds of bounded degree cannot be guaranteed to be as cache friendly as structured grids of the same size.

Our construction is based on embedding an FFT butterfly graph into a triangulation of a 3-dimensional cube. The 2^n-point FFT graph, denoted as F_n, is a graph having $(n+1)2^n = N$ vertices arranged in $n+1$ layers of 2^n vertices each, see Figure 3. In other words, vertices of F_n form an array (k,i), $0 \le k \le n, 0 \le$

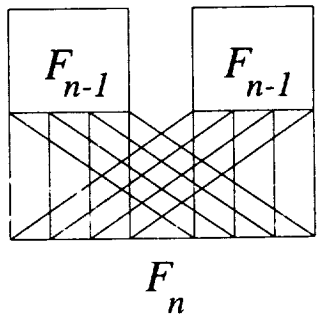

Figure 3. *A recursive construction of the FFT graph. F_n is built from two copies of F_{n-1} by adding $n+1^{th}$ layer of 2^n vertices and connecting them with vertices of n^{th} layer by the butterflies.*

$i \le 2^n - 1$ and a vertex (k,i), $k < n$ is connected with vertices $(k+1,i)$ and $(k+1, i \bigoplus 2^k)$ where $i \bigoplus 2^k$ signifies taking the complement of the k^{th} bit of i.

The FFT graph can not be tiled by sets with good surface-to-volume ratio. This can be deduced from the following inequality. For any node subset $V \subset F_n$ we have

$$|V| \le 2|\delta V| \log |\delta V| \tag{4}$$

where δV is the right boundary of V, that is, the set of points in V either on the right boundary of F_n or having a right neighbor not in V. Obviously, $|\delta V| < |V|$. Consider a partition $F_n = \bigcup V_i$, $i = 1, \ldots, k$, $|V_i| \le S$. For any subset $V \subset F_n$ it follows from (4) that $|\delta V| \ge \frac{1}{2}|V|/\log|V|$, and we can estimate the sum of

boundaries of the partition. If $S \leq 2^{n/8}$ then:

$$\sum_{i=1}^{k} |\partial(V_i)| \geq \sum_{i=1}^{k} |\delta(V_i)| - 2 \cdot 2^n \geq \frac{1}{2} \sum_{i=1}^{k} \frac{|V_i|}{\log |V_i|} - 2 \cdot 2^n$$
$$\geq \frac{1}{2 \log S} \sum_{i=1}^{k} |V_i| - 2 \cdot 2^n \geq \frac{N}{4 \log S}.$$

This estimation can be used to prove the lower bound $\Omega(M/logM)$ by the methods of [2].

Our proof of inequality (4) is based on induction and is similar to the proof of Theorem 4.1 in [5]. Let V be partitioned onto three sets A, B and C, as shown in Figure 4. From the figure we can see that

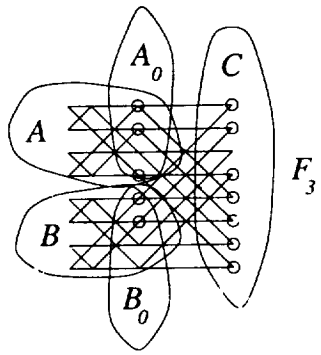

Figure 4. *Induction step for proving the surface-to-volume inequality of a subset in F_n. We can assume that $|A_0| \geq |B_0|$.*

$$|\delta V| \geq |\delta A| + |\delta B| + D + \min(0, |C| - 2|A_0|)$$

$$|V| \leq |A| + |B| + 2A_0 + \min(0, |C| - 2|A_0|)$$

where $D = |A_0| - |B_0|$. If $|C| \leq 2|A_0|$ then, by induction,

$$|V| \leq 2(|\delta A| \log |\delta A| + |\delta B| \log |\delta B| + |A_0|) \leq 2(|\delta V| \log |\delta V| - X)$$

where

$$X = |\delta A| \log(1 + \frac{|\delta B|}{|\delta A|} + \frac{D}{|\delta A|})$$
$$+ |\delta B| \log(1 + \frac{|\delta A|}{|\delta B|} + \frac{D}{|\delta B|}) + D \log(|\delta A| + |\delta B| + D) - D - B_0.$$

Since $|B_0| \leq |\delta B|$ and $|B_0| \leq |A_0| \leq |\delta A|$ and either $\log(1 + \frac{|\delta A|}{|\delta B|} + \frac{D}{|\delta B|}) \geq 1$ or $\log(1 + \frac{|\delta B|}{|\delta A|} + \frac{D}{|\delta A|}) \geq 1$, or both, then $X \geq 0$. Hence $|V| \leq 2|\delta V| \log |\delta V|$.

If $y = \min(0, |C| - 2|A_0|) > 0$ then the surface-to-volume inequality follws from the fact that $v + y \leq 2(d + y)\log(d + y)$ if $v \leq 2d \log d$.

The FFT graph can be embedded into a triangulation of a 3-dimensional cube. A recursive construction of an embedding of the FFT graph into triangulation of simplices is shown in Figure 5. The simplices can be embedded into a cube as shown in Figure 7, which than can be partitioned into parallelepipeds with further triangulation of each parallelepiped.

The butterflies connecting two last layers of F_n can be embedded into a triangulation of a simplex in such a way that the edges of the butterflies are mapped onto lines of pieces (t_0, t_3, b_7, b_4) and (t_7, t_4, b_3, b_0) of one of the ruled surfaces[1] and two skewed ruled surfaces (t_0, t_3, b_3, b_0) and (t_7, t_4, b_4, b_7). Each ruled surface separates a simplex built on the appropriate vertices onto two parts as shown in Figure 8, the top view is shown in Figure 6. The whole simplex (t_0, t_7, b_7, b_0) can be partitioned onto four above listed simplices and 5 primitive simplices: (t_3, t_4, b_3, b_4), (t_3, t_4, b_4, b_7), (t_3, t_4, b_3, b_0), (t_0, t_3, b_4, b_3) and (t_7, t_4, b_4, b_3). Each of the simplices (t_0, t_3, b_7, b_4), (t_7, t_4, b_3, b_0), (t_0, t_3, b_3, b_0) and (t_7, t_4, b_4, b_7) is separated by a ruled surface, hence it is sufficient to build a triangulation of a simplex separated by a ruled surface see Figure 8. This can be done in 3 steps: 1. adding vertices on the edges which are not parts of the ruled surface, 2. partitioning the simplex into triangular prisms and, 3. triangulating each prism.

It is easy to verify that the total number of vertices in the triangulation does not exceed $M = 3n2^n$. Hence we have constructed a triangulation having the property declared at the beginning of this section.

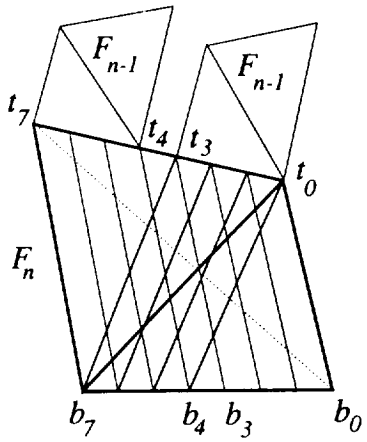

Figure 5. Recursive construction of embedding of FFT graph into a triangulations of a simplex.

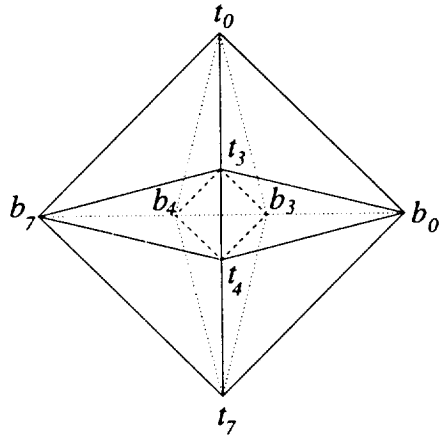

Figure 6. Embedding one layer of FFT graph into a triangulations of a simplex, top view.

[1]The ruled surfaces described here are built by linearly parametrizing two crossing lines in 3D space and connecting corresponding points by lines. A ruled surface can be viewed as a hyperboloid containing the two crossing lines.

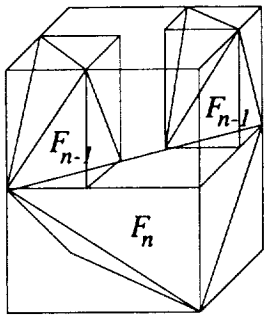

Figure 7. Recursive triangulation of a cube.

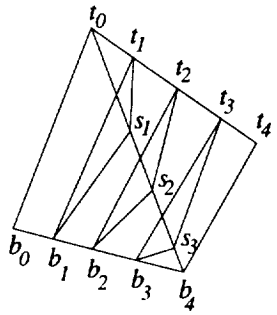

Figure 8. Triangulation of a simplex separated by a ruled surface via adding points s_1, s_2, s_3. Only partition not shadowed by the ruled surface is shown.

5 Related Work and Conclusions

The reduction of cache misses in scientific computations is an active subject of research. One of the first lower bounds for data movement between primary and secondary storage was obtained on [5]. Recently the work has focused in developing compiler techniques to reduce the number of cache misses. In this direction we mention [4], where the notion of the cache miss equation (CME) and a tiling of structured grids with conflict free rectilinear parallelepipeds were introduced. Some tight lower and upper bounds for computation of explicit operators on structured grids were obtained in [2], where a tiling with a reduced fundamental parallelepiped of the interference lattice was used for reduction of the cache misses. Some practical methods for improving cache performance in computations of explicit operators are given in [6].

We showed that the reduction of cache misses for computations of explicit local operators defined on discretization grids is closely related to the problem of covering the grids with conflict free sets having good surface-to-volume ratio. We introduced two new coverings of structured grids: a covering with Voronoi cells and a covering with rectilinear parallelepipeds built on the vectors of successive minima of the interference lattice. The cells of both coverings have near-optimal surface-to-volume ratios. Direct measurements of the cache misses show a significant advantage of the successive minima covering relative to the computations using the natural loop order, maximally optimized by a compiler. We also showed that there are bounded degree unstructured 3-dimensional grids such that any local operator on the corresponding grid has significantly larger number of cache misses than a similar operator on a structured grid. We are currently working on similar results in two dimensions.

Bibliography

[1] J.W.S. Cassels. *An Introduction to the Geometry of Numbers.* Springer-Verlag, 1997, 344 P.

[2] M. Frumkin, R.F. Van der Wijngaart. *Efficient cache use for stencil operations on structured discretization grids.* NAS Technical Report NAS-00-015, November 2000, submitted to JACM.

[3] J.L. Hennessy, D.A. Patterson. *Computer Organization and Design.* Morgan Kaufmann Publishers, San Mateo, CA, 1994.

[4] S. Gosh, M. Martonosi, S. Malik. *Cache Miss Equations: An Analytical Representation of Cache Misses.* ACM ICS 1997, pp. 317–324.

[5] J.W. Hong, H.T. Kung. *I/O Complexity: The Red-Blue Pebble Game.* IEEE Symposium on Theoretical Computer Science, 1981, pp. 326-333.

[6] G. Rivera, C.W. Tseng. *Tiling Optimizations for 3D Scientific Computations.* Proc. Supercomputing 2000, Dallas, TX, November 2000.